

1. [2 boda] Zašto je problem složenosti veliki problem pri izgradnji SRSV-a? Koji se postupci koriste radi smanjenja složenosti?

Zbog složenosti teško je osmisliti sustav, lako je unižeti greške, teško ih pronaći i ispraviti. Osnovna ideja postupaka koji se koriste je "podijeli i vladaj". Pomaže i grafički prikaz (dijela) problema.

2. [2 boda] Navesti aktivnosti procesa izgradnje programske potpore i barem dva najbitnija dijagrama UML-a za svaku aktivnost.

1. specifikacija zahtjeva
 - dijagram obrazaca uporave, sekvencijski, ...
2. oblikovanje arhitekture
 - dijagram stanja, dijagram razreda, ...
3. izgradnja
 - dijagram stanja, dijagram razreda, ...
4. ispitivanje
 - dijagram obrazaca uporave, sekvencijski, ...
5. ugradnja
 - dijagram komponenata, dijagram obrazaca uporabe, ...

3. [2 boda] Zašto se svugdje ne koristi puni operacijski sustav sa svim svojim elementima (operacijama koje omogućuje za lakšu izradu programa)?

(Pretpostavimo da se radi o operacijskom sustavu za SRSV.)

1. OS može povećati cijenu: treba više memorije, jači procesor
 2. Neke potrebne specifične operacije se možda ne mogu ostvariti ili se teško ostvaruju (možda i duže traju).
- (OS koji nije za SRSV unosi nedeterminizam, nema potrebne mehanizme za ostvarenje vremenske ispravnosti)

4. [2 boda] Navesti uporabivost neizrazite logike u kontekstu upravljanja sustavima. Koja su dobra a koja loša svojstva neizrazite logike u tom okruženju?

uporaba: za automatsko upravljanje (jednostavnijim) postupcima
dobra svojstva: logika upravljanja razumljiva čovjeku,
egzaktna funkcija upravljanja ne mora biti poznata
loša svojstva: heuristika ne daje optimalno rješenje, može biti teško podesiti parametre za željenu kvalitetu upravljanja

5. [2 boda] Što je to "najmanja gornja granica procesorskog iskorištenja" $\text{lub}(U)$? Koji se zaključci mogu donijeti za sustave za koje vrijedi $U_S \leq \text{lub}(U)$, odnosno, za one koje ista nejednakost ne vrijedi?

$U \leq \text{lub}(U)$ dovoljan uvjet rasporedivosti na jednog procesorskom sustavu kad se koristi mjera ponavljanja za dodjelu prioriteta
 $\text{lub}(U) < U \leq 1$ sustav može, ali i ne mora biti rasporediv
 $U > 1$ sustav sigurno nije rasporediv

6. Neki sustav se početno nalazi u stanju PRAZAN. Nakon prvog događaja AUTO sustav prelazi u stanje JEDAN. Drugi događaj AUTO kad je sustav u stanju JEDAN mijenja stanje sustava u DVA. Kad je sustav u stanju DVA, nakon 1 minute sustav prelazi u stanje PRIJEVOZ. Nakon 10 minuta (u stanju PRIJEVOZ) sustav prelazi u stanje ISKRCAJ. U tom stanju ostaje dok se ne dogode dva događaja ODLAZAK, kada odlazi u stanje PRAZAN. Pretpostaviti da se događaj AUTO neće dogoditi dok sustav nije u stanju PRAZAN ili JEDAN te da će se događaj ODLAZAK dogoditi samo kada je sustav u stanju ISKRCAJ (i to najviše dva takva događaja).

a) [3 boda] Opisati sustav sekvencijskim dijagramom UML-a za slučaj kada se događaji AUTO događaju u 5. i 15. minuti te događaji ODLAZAK u 30. i 40. minuti.

b) [3 boda] Prikazati sustav vremenskom Petrijevom mrežom.

7. [3 boda] Neki sustav treba upravljati s tri aktivnosti A, B i C. Aktivnost A je periodička i jednom unutar svakih 20 ms treba pozvati funkciju $A()$. Aktivnost B je također periodička, ali ju treba pozivati (funkcijom $B()$) svakih 50 ms i obrada mora biti gotova unutar 10 ms (ona se smije izvoditi u intervalima: prvi puta od 0. do 10. ms, drugi puta od 50. do 60. treći puta od 100. do 110. i tako dalje). Aktivnost C je periodička i jednom unutar svakih 100 ms treba pozvati funkciju $C()$. Maksimalno trajanje svake aktivnosti ($A()$, $B()$, $C()$) je 5 ms. Jednom započeta aktivnost se mora izvesti do kraja. Opisati ostvarenje upravljanja ako sustav posjeduje samo sučelje $dohvati_sat()$ koje vraća vrijeme u milisekundama.

```
T_A = 20, T_B = 50, T_C = 100
t_A = t_B = t_C = 0
ponavljanje { //najviše jedan zadatak u petlji !
    ako je dohvati_sat() >= t_B tada
        B()
        t_B += T_B
    inače ako je dohvati_sat() >= t_A tada
        A()
        t_A += T_A
    inače ako je dohvati_sat() >= t_C tada
        C()
        t_C += T_C
}
```

8. [2 boda] Neki PID regulator zadan je parametrima $K_P = 0,5$, $K_I = 0,2$ i $K_D = 0,05$. Ako je stanje sustava u nekom trenutku $y_k = 100$, željeno stanje $y_P = 200$ izračunati r_k (vrijednost s kojom se utječe na sustav)? Prethodno stanje sustava je $y_{k-1} = 80$, uz $I_{k-1} = 20$. Korak integracije je $T = 0,1$ s.

9. [3 boda] Neka automatska miješalica betona opremljena je senzorom viskoznosti. U miješalicu treba staviti pijeska (koji može imati različiti postotak vlage), cement i vode. Inicijalno se u miješalicu doda željena količina pijeska, te inicijalna količina cementa i vode, a nakon toga potrebno je s manjim dodavanjem cementa i vode postići željenu viskoznost: kada je viskoznost veća od željene V_P treba dodati vode, a kada je manja treba dodati cementa. Upravljanje tim dodatnim dodavanjima ostvariti korištenjem neizrazite logike. Pretpostaviti da upravljački sustav za izlazne vrijednosti iz neizrazitog upravljača manje od nule (do -1) dodaje vode (što je vrijednost bliža -1 pušta se više vode), a za vrijednosti veće od nule (do 1) dodaje se cement (više cementa za veće vrijednosti). Pokazati rad sustava na primjeru kada je trenutna viskoznost nešto veća od željene.

10. Zadan je sustav zadataka $\mathcal{S}_1$ koji se raspoređuje prema mjeri ponavljanja (engl. *rate monotonic priority assignment – RMPA*):

$$\mathcal{T}_1 : T_1 = 10 \text{ ms}, C_1 = 3 \text{ ms}$$

$$\mathcal{T}_2 : T_2 = 15 \text{ ms}, C_2 = 5 \text{ ms}$$

$$\mathcal{T}_3 : T_3 = 20 \text{ ms}, C_3 = 4 \text{ ms}$$

$$\mathcal{T}_4 : T_4 = 30 \text{ ms}, C_4 = 2 \text{ ms}$$

- a) [2 boda] Grafičkim postupkom provjeriti rasporedivost na jednoprocesorskom sustavu.
- b) [2 boda] Korištenjem općeg kriterija provjeriti je li sustav rasporediv.
- c) [1 bod] Odrediti implicitni trenutak krajnjeg dovršetka za sve zadatke. Nije neophodno koristiti formule.